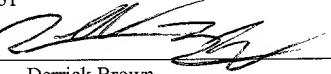


**PATENT  
5181-78600  
P5415**

"EXPRESS MAIL" MAILING LABEL  
NUMBER EL726369517US

DATE OF DEPOSIT 1/25/01

I HEREBY CERTIFY THAT THIS PAPER OR  
FEE IS BEING DEPOSITED WITH THE  
UNITED STATES POSTAL SERVICE  
"EXPRESS MAIL POST OFFICE TO  
ADDRESSEE" SERVICE UNDER 37 C.F.R. §  
1.10 ON THE DATE INDICATED ABOVE  
AND IS ADDRESSED TO THE ASSISTANT  
COMMISSIONER FOR PATENTS, BOX  
PATENT APPLICATION, WASHINGTON,  
D.C. 20231



Derrick Brown

System and Method for Storing and Retrieving  
Bookmark Information

By:

Trung M. Tran

Atty. Dkt. No.: 5181-78600

B. Noël Kivlin/JLB  
P.O. Box 398  
Austin, TX 78767-0398  
Ph: (512) 476-1400

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates to the field of application software, and more particularly to a system and method for enabling an application to store bookmark information on and retrieve bookmark information from a server computer, wherein the bookmark information specifies one or more uniform resource locators (URLs).

### 2. Description of the Related Art

One development that has encouraged the proliferation of inter-networked computer systems and applications has been the development of a standard technique for identifying and accessing network resources. In particular, the standardization of the uniform resource locator (URL) has been important. Uniform resource locators are commonly used to identify virtually any type of resource, including files, executable components, multimedia resources, etc.

For example, a URL can be used to reference a file according to the following format:

protocol://host.domain[:port]/path/filename

The “protocol://” portion specifies a communication protocol to be used in accessing the file. Commonly used protocols include: “http://” (HyperText Transfer Protocol), “ftp://” (File Transfer Protocol), and “file://” (local file system). The “host.domain” portion is the Internet address of the site where the file is located. The “[port]” portion identifies a port number on a server computer used to access the file. The “/path/” portion specifies where in a directory tree the file is located. The “filename” portion is the name of the file.

A large variety of software applications utilize URLs to reference resources. Applications such as Microsoft Internet Explorer and Netscape Navigator utilize URLs to reference resources on the World Wide Web and are commonly referred to as “web browsers”. However, any of various types of applications, such as word processors,

multimedia applications, etc., may utilize URLs to reference resources located on the World Wide Web or elsewhere. As used herein, an application said to have web browsing functionality may refer to any of various applications operable to utilize URLs to reference network resources.

5       Web browsers and other applications with web-browsing functionality often maintain a list of “bookmarks,” wherein each bookmark specifies a URL. For example, bookmarks are accessible in the Microsoft Internet Explorer application via a “Favorites” menu and are accessible in the Netscape Navigator application via a “Bookmarks” menu. Bookmarks are convenient for users since they provide an easy and quick way to access a  
10      URL.

As networked systems and applications penetrate more and more into everyday life, users often find themselves using multiple computer systems. For example, users may use computer systems that are not their own personal computer system. For example, users now commonly use public computer systems located in libraries, cybercafes, etc. Also,  
15      users often use multiple computer systems at home and at work. In these situations, a user often desires to access a URL which he had previously created a bookmark for on another computer system. However, the prior art provides no techniques for accomplishing this.

In the prior art, creating a bookmark results in bookmark information being saved locally, that is, on the host computer system that is executing the current application the  
20      user is using. However, when the user moves to a different computer system, there is no way for the different computer system to access the previously created bookmark. Thus, in the current conception of a bookmark, bookmarks are associated with a particular computer system. It would be desirable to provide a method wherein the concept of a  
25      bookmark is such that bookmarks are associated with users and are independent of any particular computer system. It would also be desirable to provide a method for users to access a set of bookmarks from multiple computer systems. It would also be desirable for the method to be tightly integrated into a user application operable to store, retrieve, and use bookmark information.

## **SUMMARY OF THE INVENTION**

The problems outlined above may in large part be solved by providing a system and method for enabling bookmark information to be shared among different computer systems, such as described herein. A method for storing bookmark information is described.

5 According to one embodiment of this method, user input requesting to store bookmark information may be received, wherein the bookmark information specifies a uniform resource locator (URL) and possibly also specifies other information, such as a name to associate with the URL. In one embodiment, the method may be operable to store the bookmark information either locally or remotely, depending on what the user desires. If

10 the user wants to store the bookmark information locally then the bookmark information may be stored on the host computer system that is executing a current application the user is using. If the user does not want to store the bookmark information locally, then the bookmark information may be stored on a server computer system.

If the bookmark information is to be stored on the server computer system, then

15 the method may prompt for and receive user input specifying user authentication information, e.g., a username and password. It may be desirable to store the bookmark information on the server computer system in association with the particular user. For example, bookmark information may be stored on the server computer system for multiple users, wherein the bookmark information for each user specifies a set of URLs.

20 Thus, the user authentication information may be used in associating the bookmark information with the correct user. The user authentication information may also be used in verifying the identity of the user, to prevent unauthorized users from adding bookmark information for another user.

After authenticating the user using the user authentication information, the

25 method may operate to send the bookmark information to the server computer system. The server computer system may store the bookmark information, in response to receiving the bookmark information. As described above, the server computer system may store the bookmark information in association with the particular user. Thus, the method may also send user identification information to the server computer system that

the server computer system can use to determine the user with which to associate the bookmark information. Storing the bookmark information may comprise adding the bookmark information to existing bookmark information that is already stored for the user.

5       The bookmark information may be sent to the server computer system in any of various ways. For example, any of various communication protocols may be used to interface with the server computer system. In one embodiment, the Lightweight Directory Access Protocol (LDAP) may be used. In other embodiments, other protocols may be used. For example, a custom TCP and/or IP-based protocol may be  
10      defined and implemented.

In one embodiment, an application programming interface (API) may be defined, which enables client applications to communicate with the server computer system to store and/or retrieve bookmark information. For example, the API may include function or method calls for storing bookmark information remotely, retrieving a list of remotely  
15      stored bookmarks associated with a particular user, etc. Such an API may enable various types of client applications to easily incorporate the functionality of storing bookmark information remotely and retrieving remotely stored bookmark information. For example, an executable module implementing the API may be created, such as a DLL, shared library, Java module, etc. The client applications may then call the executable  
20      module to perform the desired function.

For example, an application having web-browsing functionality may call the executable module via the API to obtain a list of bookmark information stored for a certain user in the server computer system. After obtaining the bookmark information, the application may then enable the user to access the bookmark information via a  
25      graphical user interface. For example, for each URL specified by the bookmark information, the URL may be placed in a menu of the application, possibly identified by a name also specified by the bookmark information, such that the user can select the desired URL from the menu, e.g., in order to display or otherwise access the resource referenced by the URL.

A method for obtaining stored bookmark information is also described. According to one embodiment of the method, user input requesting to retrieve stored bookmark information may be received. For example, this user input may be received via a graphical user interface, e.g., when the user selects a “Get My Bookmarks” menu command. In an alternative embodiment, this step may not be necessary. For example, the method may operate to automatically obtain bookmark information, e.g., at application startup time.

In response, the method may prompt for and receive user input specifying user authentication information, e.g., a username and password. The method may interface with a server computer system to authenticate the user, using the user authentication information. As described above, bookmark information stored on the server computer system may be associated with a particular user, and the user authentication may be performed in order to identify the user (identify the appropriate bookmark information to obtain), prevent unauthorized users from obtaining bookmark information for another user, etc.

Bookmark information may then be received from the server computer system. The bookmark information received may be bookmark information that is associated with the user, e.g., as determined by the user authentication information. The bookmark information may specify a plurality of URLs which were previously stored on the server computer system in accordance with the method described above. The bookmark information may also comprise other information, such as a name that is associated with each URL.

Any of various communication protocols may be utilized in retrieving the bookmark information. In one embodiment, the Lightweight Directory Access Protocol (LDAP) may be used. In other embodiments, other protocols may be used. For example, a custom TCP and/or IP-based protocol may be defined and implemented. Also, as described above, an application may call an API to request an executable module to

interface with the server computer and may receive the bookmark information from the call to the executable module.

After receiving the bookmark information, the method may enable the user to access the bookmark information via a graphical user interface. As described above, in 5 one embodiment this may comprise placing each URL specified by the bookmark information in a menu such that the user can select the desired URL from the menu, e.g., in order to display or otherwise access the resource referenced by the URL. If bookmark information is organized hierarchically, then the URLs may be nested in the menu according to the appropriate hierarchy.

10

The methods described above for storing bookmark information and obtaining stored bookmark information may be utilized to enable a method for sharing bookmark information among different computer systems. A first computer system may execute a first software application having web-browsing functionality. The first software 15 application having web-browsing functionality may interface with a server computer system in order to remotely store bookmark information on the server computer system, in accordance with the method described above.

A second computer system may execute a second software application having web-browsing functionality. The second software application having web-browsing 20 functionality may interface with the server computer system in order to retrieve the stored bookmark information, in accordance with the method described above. The second software application having web-browsing functionality may then enable the user to access the retrieved bookmark information via a graphical user interface such as a menu, as described above.

25 Thus, the method may enable a user to use a common set of bookmarks from the context of different computer systems. When at a first computer system, the user can request to store a bookmark on a server computer system, much the same as the user requests to store a bookmark locally using current web browser applications, with a possible additional step of also specifying user authentication information. After moving

to a second computer system, an application on the second computer system may then retrieve the stored bookmark information from the server computer system. The bookmark information may then be tightly integrated into the graphical user interface of the application, such that the user can easily access the bookmark information, e.g., much  
5 the same as bookmarks are accessed in current web browser applications.

0 0 0 0 0 0 0 0 0 0

## **BRIEF DESCRIPTION OF THE DRAWINGS**

A better understanding of the present invention can be obtained when the following detailed description is considered in conjunction with the following drawings,  
5 in which:

Figure 1 illustrates one embodiment of a system including a client computer system connected via a network to a server computer system;

10 Figure 2 is a block diagram illustrating one embodiment of the client computer system shown in Figure 1;

Figure 3 is a flowchart diagram illustrating one embodiment of a method for  
storing bookmark information;

15 Figure 4 is a flowchart diagram illustrating one embodiment of a method for  
obtaining bookmark information; and

Figure 5 is a flowchart diagram illustrating one embodiment of a method for  
20 sharing bookmark information among different computer systems.

While the invention is susceptible to various modifications and alternative forms,  
specific embodiments thereof are shown by way of example in the drawings and are  
herein described in detail. It should be understood, however, that the drawings and  
25 detailed description thereto are not intended to limit the invention to the particular form  
disclosed, but on the contrary, the intention is to cover all modifications, equivalents and  
alternatives falling within the spirit and scope of the present invention as defined by the  
appended claims.

## **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

### **Figure 1 – Client Computer System Connected to a Server Computer System**

Figure 1 illustrates one embodiment of a system including a client computer system 82 connected via a network 84 to a server computer system 86. The client computer system 82 may be a computer system of any type. Defined broadly, a computer system may refer to any system operable to execute program instructions from a memory medium. As examples, in various embodiments, the client computer system 82 may comprise a personal computer, a workstation, a consumer device such as a cellular telephone or personal digital assistant (PDA), a network appliance, etc.

The client computer system 82 may be connected to a network 84. The network 84 may be any type of network, including the Internet, a LAN, or a WAN, among others. As shown, a server computer system 86 may also be connected to the network 84. The server computer system 86 may be a computer system of any type. The network 84 enables the client and server computer systems to communicate with each other. For example, in one embodiment, the network 84 may be the Internet, and the client computer system 82 may communicate with the server computer system 86 using a TCP and/or IP-based protocol.

The client computer system 82 may execute software operable to reference network resources using uniform resource locators (URLs). For example, the client computer system 82 may execute an application having web-browsing functionality that allows users to access, e.g., view, various resources referenced by URLs. The software executed by the client computer system 82 may support the use of bookmarks. Each bookmark may specify a uniform resource locator (URL). In one embodiment, a bookmark may also comprise other information. For example, the URL of a bookmark may be associated with a name. For example, the software may enable users to select from bookmarks via a graphical user interface that displays the name of each bookmark, such as a menu, in order to access the bookmarks, e.g., to access resources referenced by the URLs of the bookmarks.

The client computer system 82 may be operable to retrieve bookmark information from the server computer system 86, wherein the bookmark information specifies one or more URLs. One embodiment of a method for obtaining bookmark information is described below. The client computer system 82 may also be operable to store bookmark information on the server computer system 86. One embodiment of a method for storing bookmark information is described below. As described below, the bookmark information may be stored in association with a particular user. For example, this may allow multiple users to keep their own sets of bookmarks stored on the server computer system.

As shown, the server computer system 86 may utilize a database 90 to store bookmark information received from the client computer system 82. It is noted that Figure 1 is exemplary only, and the server computer system 86 and/or database 90 may be arbitrarily complex. For example, in order to support a large number of interactions with many client computer systems simultaneously, the server computer system 86 may comprise multiple server computers and/or databases.

The ability to store bookmark information on the server computer system and later retrieve the stored bookmark information may benefit users by allowing them to work with their bookmarks on multiple computer systems. For example, a user of a first client computer system may desire to store bookmark information on the server computer system, wherein the bookmark information specifies a URL. For example, the user may request an application operable to use the URL to store the bookmark information on the server computer system. If the user later moves to a second client computer system, the user may then desire to retrieve the previously stored bookmark information from the server computer system. For example, the user may request an application operable to use the URL to retrieve the bookmark information from the server computer system. The application may then enable the user to access the retrieved bookmark information via a graphical user interface.

Figure 2 – Client Computer System Block Diagram

Figure 2 is a block diagram illustrating one embodiment of the client computer system 82 shown in Figure 1. It is noted that any type of computer system configuration or architecture can be used as desired, and Figure 2 illustrates a representative PC embodiment. It is also noted that the client computer system 82 may comprise a general purpose computer system, may comprise a consumer device such as a cellular telephone or personal digital assistant (PDA), or may comprise any other type of computer system. Elements of a computer not necessary to understand the present disclosure have been omitted for simplicity.

10       The computer may include at least one central processing unit or CPU 160 which is coupled to a processor or host bus 162. The CPU 160 may be any of various types, including an x86 processor, e.g., a Pentium class, a PowerPC processor, a CPU from the SPARC family of RISC processors, as well as others. Main memory 166 is coupled to the host bus 162 by means of memory controller 164.

15       The host bus 162 is coupled to an expansion or input/output bus 170 by means of a bus controller 168 or bus bridge logic. The expansion bus 170 may be any type of bus, including the PCI (Peripheral Component Interconnect) expansion bus. The expansion bus 170 includes slots for various devices such as the sound card 122 and the modem card 124. The computer further includes a video display subsystem 180 and hard drive 182  
20       coupled to the expansion bus 170.

25       The main memory 166 may store operating system software as well as other software for operation of the computer system. The main memory 166 may also store a computer program operable to perform one or more of the methods described below for storing bookmark information on a server computer system and/or retrieving bookmark information from the server computer system.

A computer program operable to perform the methods described below may also be stored on other types of memory media. The term “memory medium” is intended to include an installation medium, e.g., a CD-ROM, floppy disks, or tape device, a computer system memory or random access memory such as DRAM, SRAM, EDO RAM, etc., or a

non-volatile memory such as a magnetic media, e.g., a hard drive, or optical storage. The memory medium may comprise other types of memory as well, or combinations thereof. In addition, the memory medium may be located in a first computer in which the programs are executed, or may be located in a second different computer which connects to the first computer over a network, such as the Internet. In the latter instance, the second computer may provide the program instructions to the first computer for execution.

Figure 3 – Storing Bookmark Information

10       Figure 3 is a flowchart diagram illustrating one embodiment of a method for storing bookmark information. In one embodiment, the method of Figure 3 may be performed by the client computer system 82 illustrated in Figure 1. For example, the method may be implemented by an application having web-browsing functionality that executes on the client computer system 82.

15       In step 200, user input requesting to store bookmark information may be received, wherein the bookmark information specifies a uniform resource locator (URL). For example, the user may request via a graphical user interface to bookmark a URL, e.g., a URL referencing a resource currently being displayed or otherwise accessed. The user may also specify other information to be associated with the URL. For example, in  
20 response to a user request to store bookmark information, the method may present a window or dialog box enabling the user to specify a name to associate with the URL. For example, the URL may be “<http://www.sun.com>”, and the user may specify a name of “Sun Microsystems”. If bookmark information is categorized hierarchically, the user may also specify a subfolder or other hierarchical location for the new bookmark  
25 information.

      In one embodiment, the method may be operable to store the bookmark information either locally or remotely, depending on what the user desires. Thus, in step 202, the method may determine whether the user wants to store the bookmark information on a server computer system, such as the sever computer system 86 discussed

above with reference to Figure 1. For example, the dialog box or window that enables the user to specify the bookmark name in step 200 may also enable the user to specify whether to store the bookmark information locally or remotely. Alternatively, there may be separate user commands for storing bookmark information locally and storing  
5 bookmark information remotely.

If the user does not want to store the bookmark information on the server computer system, then in step 204, the bookmark information may be stored locally, that is, on the host computer system that is executing the current application the user is using. The bookmark information may be stored locally according to prior art techniques.

10 If the user does want to store the bookmark information on the server computer system, then in step 206, the method may prompt for and receive user input specifying user authentication information. It may be desirable to store the bookmark information on the server computer system in association with the particular user. For example, bookmark information may be stored on the server computer system for multiple users,  
15 wherein the bookmark information for each user specifies a set of URLs. Thus, the user authentication information may be used in associating the bookmark information with the correct user.

The user authentication information may also be used in verifying the identity of the user, to prevent unauthorized users from adding bookmark information for another  
20 user. In various embodiments, the user authentication information may comprise any type of information usable to perform an authentication procedure to authenticate the user, such as a username and password, biometric information, etc. Thus, in various embodiments, the method may prompt for and receive the user authentication information in various ways, e.g., by displaying a dialog box for entering a username and password,  
25 by receiving fingerprint information via a hardware device, etc.

In step 208, the method may interface with the server computer system in order to authenticate/authorize the user. For example, in one embodiment, authenticating the user may comprise verifying that the password provided correctly matches the username provided.

If the user has been authenticated previously, then user information and authentication information may have been stored, making it unnecessary to perform steps 206 and 208. For example, the user may have previously requested to store bookmark information on the server computer system, or the user may have been prompted for user 5 information and authentication information at application startup time. Also, if the user previously entered user authentication information in order to obtain existing stored bookmark information, as described below with reference to Figure 4, then the user may not need to re-enter the information.

If the user cannot be authenticated, an error may be indicated. Otherwise, in step 10 210, the method may operate to send the bookmark information to the server computer system. In step 212, the server computer system may store the bookmark information, in response to receiving the bookmark information. As illustrated in Figure 1, the bookmark information may be stored in a database 90. As described above, the server computer system may store the bookmark information in association with the particular user. Thus, 15 in step 210, the method may also send user identification information to the server computer system that the server computer system can use to determine the user with which to associate the bookmark information. Storing the bookmark information may comprise adding the bookmark information to existing bookmark information that is already stored for the user.

20 In various embodiments, the server computer system may implement various storage management rules for increasing storage efficiency. For example, if bookmark information is stored for a large number of users and a limited amount of storage space is available, it may be desirable to limit the amount of time that bookmark information is stored. For example, if a user does not access his bookmark information for a period of 25 six months, the server computer system may automatically delete the bookmark information. Also, users may sometimes forget that certain URLs are already bookmarked. Thus, when bookmark information is received, the server computer system may be operable to check to see whether the bookmark information is already stored, and

if so, the bookmark information may not be re-stored. The server computer system may send a message to this effect to be indicated to the user.

The method may interface with the server computer system to send the bookmark information to the server computer system in any of various ways. For example, any of 5 various communication protocols may be used to interface with the server computer system. In one embodiment, the Lightweight Directory Access Protocol (LDAP) may be used. In this case, the server computer system may be operable to store the bookmark information using standard LDAP techniques, and thus if the server computer system supports LDAP, then no special modifications may be required to enable the storage of 10 bookmark information.

In other embodiments, other protocols may be used. For example, a custom TCP and/or IP-based protocol may be defined and implemented. For example, a TCP/IP protocol may be operable to marshal bookmark information into a buffer, stream the buffer to the server computer system, receive responses from the server computer system, 15 etc.

In one embodiment, an application programming interface (API) may be defined, which enables client programs to communicate with the server computer system to store and/or retrieve bookmark information. For example, the API may include function or method calls for storing bookmark information remotely, retrieving a list of remotely 20 stored bookmarks associated with a particular user, etc. Such an API may enable various types of client applications to easily incorporate the functionality of storing bookmark information remotely and retrieving remotely stored bookmark information. For example, an executable module implementing the API may be created, such as a DLL, shared library, Java module, etc. The client applications may then call the executable 25 module to perform the desired function.

For example, an application having web-browsing functionality may call the executable module via the API to obtain a list of bookmark information stored for a certain user in the server computer system. After obtaining the bookmark information, the application may then enable the user to access the bookmark information via a

graphical user interface. For example, for each URL specified by the bookmark information, the URL may be placed in a menu of the application, possibly identified by a name also specified by the bookmark information, such that the user can select the desired URL from the menu, e.g., in order to display or otherwise access the resource referenced by the URL.

5

Figure 4 – Obtaining Bookmark Information

Figure 4 is a flowchart diagram illustrating one embodiment of a method for obtaining bookmark information. In one embodiment, the method of Figure 4 may be performed by the client computer system 82 illustrated in Figure 1. For example, the method may be implemented by an application having web-browsing functionality that executes on the client computer system 82.

In step 300, user input requesting to retrieve stored bookmark information may be received. For example, this user input may be received via a graphical user interface, e.g., when the user selects a “Get My Bookmarks” menu command. In an alternative embodiment, step 300 may not be necessary. For example, the method may operate to automatically obtain bookmark information, e.g., at application startup time.

In step 302, the method may prompt for and receive user input specifying user authentication information. Similarly as described above with reference to Figure 3, any of various types of user authentication information may be received. In step 304, the method may interface with a server computer system to authenticate/authorize the user, using the user authentication information. As described above, bookmark information stored on a server computer system may be associated with a particular user, and steps 20 302 and 304 may be performed in order to identify the user (identify the appropriate bookmark information to obtain), prevent unauthorized users from obtaining bookmark information for another user, etc.

In step 306, bookmark information may be received from the server computer system. The bookmark information received may be bookmark information that is

associated with the user, e.g., as determined by the user authentication information received in step 302. The bookmark information may specify a plurality of URLs which were previously stored on the server computer system in accordance with the method of Figure 3. As described above, the bookmark information may also comprise other information, such as a name that is associated with each URL.

Similarly as described above with reference to Figure 3, any of various communication protocols may be utilized in step 306. In one embodiment, the Lightweight Directory Access Protocol (LDAP) may be used. For example, the user authentication information received in step 302 may comprise an LDAP common and/or unique name for the user, and the method may request to retrieve a list of bookmarks stored on the server computer system under that user's LDAP name.

In other embodiments, other protocols may be used. For example, a custom TCP and/or IP-based protocol may be defined and implemented. For example, a TCP/IP protocol may be operable to request the bookmark information from the server computer system, receive the bookmark information from the server computer as a marshaled buffer, etc. Also, as described above, an application may call an API to request an executable module to interface with the server computer and may receive the bookmark information from the call to the executable module.

In step 308, the method may enable the user to access the bookmark information received from the server computer system via a graphical user interface. As described above, in one embodiment this may comprise placing each URL specified by the bookmark information in a menu such that the user can select the desired URL from the menu, e.g., in order to display or otherwise access the resource referenced by the URL. If bookmark information is organized hierarchically, then the URLs may be nested in the menu according to the appropriate hierarchy.

Figure 5 – Sharing Bookmark Information Among Different Computer Systems

Figure 5 is a flowchart diagram illustrating one embodiment of a method for sharing bookmark information among different computer systems.

5 In step 400, a first computer system may execute a first software application having web-browsing functionality. The first software application may be operable to utilize URLs in order to access resources for any of various purposes.

10 In step 402, the first software application having web-browsing functionality may interface with a server computer system in order to remotely store bookmark information on the server computer system. The method of Figure 3 may be used to perform step 402.

In step 404, a second computer system may execute a second software application having web-browsing functionality. The second software application may be operable to utilize URLs in order to access resources for any of various purposes.

15 In step 406, the second software application having web-browsing functionality may interface with the server computer system in order to retrieve the stored bookmark information. The method of Figure 4 may be used to perform step 406.

In step 408, the second software application having web-browsing functionality may enable the user to access the retrieved bookmark information via a graphical user interface such as a menu, as described above.

20 Thus, the method of Figure 5 may enable a user to use a common set of bookmarks from the context of different computer systems. When at a first computer system, the user can request to store a bookmark on a server computer system, much the same as the user requests to store a bookmark locally using current web browser applications, with a possible additional step of also specifying user authentication information. After moving to a second computer system, an application on the second computer system may then retrieve the stored bookmark information from the server computer system. The bookmark information may then be tightly integrated into the graphical user interface of the application, such that the user can easily access the

bookmark information, e.g., much the same as bookmarks are accessed in current web browser applications.

Although the embodiments above have been described in considerable detail,  
5 numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

PROPRIETARY MATERIAL  
© 2010